Performance Test Results Report for the Sled player

The Open University

Created: 17th April 2007

Author Simon Hutchinson <s.h.hutchinson@open.ac.uk>

Cross References None

Revision History

Version	Date	Author	Reason
1.0	17th Apr 2007	Simon Hutchinson	Initial Version

File Sled Performance Testing Results.pdf

Table of Contents

Table of Contents *3*
Summary of results **4**
Scope **5**
Purpose **5**
Testing Application **5**
Test Description **5**
Performance Acceptance Criteria *6*
Introduction **6**
Performance Criteria **6**
Objectives 6
Engagement Complete Criteria **6**
System Architecture **7**
Baseline Test *8*
Introduction **8**
Baseline Results 9
Benchmarking *10*
Introduction **10**
Benchmark Results 20 users 11
Other scheduled test results *12*
Scheduled Tests **12**
User Experience Tests **12**
User Experience Test Results **12**
50 User load 13
100 User load 14
150 User load (1.5X the acceptance load) 15
200 User load (2X the acceptance load) 16
Stability Tests **17**
Stress test 17
Stress Test Results 18
Conclusions and recommendations *20*
Consolidated Results **20**
Conclusions **20**
Recommendations ..... Error! Bookmark not defined.
Appendix *21*

Summary of results

All criteria were met:

The Performance Testing effort was completed 14th April 2007. The Performance Tester discovered that on the hardware and software configuration tested, that a 100 concurrent user load produced an acceptable user experience within the guidelines of a 7 second response time range 95% of the time.

Scope

This document includes a summary of performance testing results and conclusions/recommendations for the Sled application. This document does not address functional testing, nor does it address detailed application tuning.

Purpose

The purpose of this document is to report on the Sled actual performance as compared to the acceptance criteria enumerated in the Performance Testing Strategy section below. Specifically, this document details the:

- Performance Acceptance Criteria
- Measurements gathered for the application
- Summary of tests performed
- Summary of measurements collected
- Results and conclusion
- Appendices for supporting data

Testing Application

The tool used for the performance testing was Open Systems Testing Architecture (OpenSTA). This tool was selected for a number of reasons amongst which the most important were.

1. Freely available under the GNU GPL (General Public License)
2. The ability to record a web session.
3. APIs (Script Control Language) for fine grained control over test scripts.
4. A full suite of test reports.
5. A community portal and mailing list.

Test Description

The test utilized the “Developing Multimedia” UoL, which was both developed at and is in use at Liverpool Hope University. The UoL is at Learning design Level B.

A user session was recorded which completed each activity in weeks one and two of the UoL.

A constant wait time of 30 seconds was then applied between each request. To ensure that responses were being returned accurately during the test the script was amended to parse the response text for an expected string and to log a “success” or “failure” message based on its presence.

For each test (See test description for each test) a number of virtual users were assigned at varying schedules. Each virtual user is a representation of a single user’s browser session, complete with cookies for session handling. The virtual users run concurrently but are started at different times. Each virtual user completes a number of iterations of the scripted browser session.

The system was setup with a single run of the UoL and 250 users were assigned its “Learner” role.

The JBoss server was restarted between tests.

Performance Acceptance Criteria

Introduction

Performance efforts always have two sets of criteria associated with them. The first are performance criteria (requirements and goals), and the second are engagement completion criteria. In the sections below, both types of criteria are explained in general and in specific detail for the Sled performance testing effort. The performance effort will be deemed complete when either all of the performance criteria are met, or any one of the engagement completion criteria is met.

Performance Criteria

Performance criteria are the specific target performance requirements and goals of the system under test. The preferred result is that the application meets all of these goals and requirements currently and/or tunes the application until these goals are met. If this is not possible, at least one of the engagement criteria from the next section must be met for overall performance acceptance.

Objectives

The objectives of the Performance Testing Effort are:

- To validate the scalability and operability of the technical architecture on a shared platform (up to 100 concurrent users)
- To validate system performance of :
    - All user actions that require a page or screen to be loaded or refreshed will be fully displayed in 7 seconds 95% of the time when accessed over a 10Mbs LAN while there is a 100 user load on the system.
    - To validate that the system does not exhibit any critical failures under stress (unrealistic load)
    - Identify and ensure that performance issues uncovered outside of the stated performance criteria are documented.

Engagement Complete Criteria

In cases where performance requirements or goals cannot be achieved due to situations outside of the control of the Performance Testing Team, the performance effort will be considered complete when any of the following conditions are met:

- All bottle necks preventing the application from achieving the performance criteria are determined to be outside Performance Testers control
- The Performance Tester and stakeholders agree that the application performs acceptably, although some performance requirements or goals have not been achieved.

System Architecture

A standard system architecture as detailed below has been used for all tests.

Test Client (Running OpenSTA to simulate Virtual Users)

Operating System	Windows XP Professional Service Pack 2 (build 2600)
Processor	3.00 gigahertz Intel Pentium 4 8 kilobyte primary memory cache 512 kilobyte secondary memory cache
Main Circuit Board	Board: Intel Corporation D865GLC AAC27499-407 Serial Number: BTLC40912203 Bus Clock: 200 megahertz BIOS: Intel Corp. BF86510A.86A.0069.P21.0409290100 09/29/2004
Memory	496 Megabytes Installed Memory
Communications	Intel(R) PRO/100 VE Network Connection
Relevant Software & Configuration	OpenSTA 1.4.3.20

Web server / App server

Operating System	Windows Server 2003 Standard Edition Service Pack 1 (build 3790)
Processor	3.00 gigahertz Intel Pentium 4 16 kilobyte primary memory cache 1024 kilobyte secondary memory cache
Main Circuit Board	Board: Intel Corporation D865GLC AAC27499-410 Serial Number: BTLC51600823 Bus Clock: 200 megahertz BIOS: Intel Corp. BF86510A.86A.0071.P22.0411242209 11/24/2004
Memory	1008 Megabytes Installed Memory
Communications	Intel(R) PRO/100 VE Network Connection
Relevant Software & Configuration	Java version: JRE 1.5.0_11-b03 [options for JBoss] -Xms128m -Xmx512m Application Server: JBoss 4.0.4.GA Web server: Tomcat 5.5.17 (version shipped with JBoss)

Baseline Test

Introduction

Baseline results represent each user activity being performed by a single user over multiple iterations. These baselines were used primarily to validate that the scripts have been developed correctly. All baselines were executed a minimum of 30 times. All reported times are statistical calculations (averages) of all 30 iterations. The user wait time (the time between user interactions with the system) was exactly 30 seconds to ensure baseline tests are identical.

Baseline Results

Transaction	Avg Time (sec)	Std Dev Time (sec)	95th Percentile Time (sec)
LOGIN	0.0	0.1	0.0
LOGIN_COURSE	0.1	0.1	0.1
ACTIVITY_MESSAGE_FACILITY	0.2	0.1	0.2
ACTIVITY_MESSAGE	0.2	0.0	0.2
SET_PROPERTY_SEND_MESSAGE	0.2	0.0	0.2
COMPLETE_ACTIVITY_VIEW_SAMPLE_MM_PROJ	0.1	0.0	0.2
ACTIVITY_UPLOAD_ACTIVITY	0.2	0.0	0.2
QTI_POST_SELECT_TITLE	0.2	0.0	0.2
ACTIVITY_COMPLETE_AND_UPLOAD_DIR_ACT	0.1	0.0	0.2

Table 1 – Baseline Results

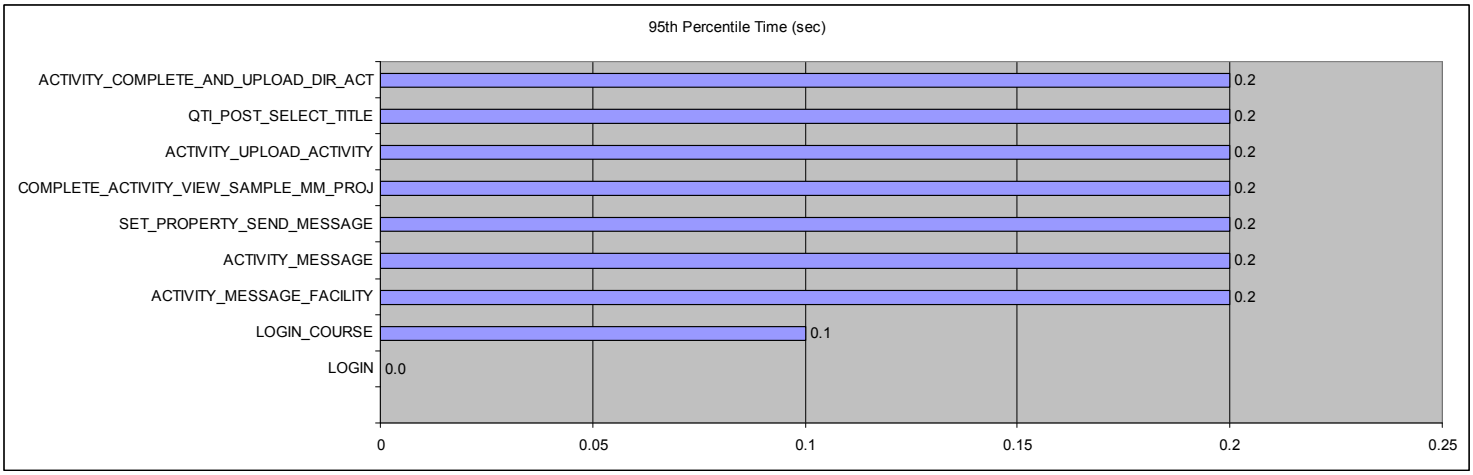

Figure 1 Baseline chart

Benchmarking

Introduction

A benchmark, or light load, scenario is generally a small community of users compared to the target load. This community of users must be large enough to represent a reasonable sample of the entire user community. Executing these tests ensured that the testing environment behaves as expected under light load before more demanding testing begins.

Additionally, the results of these tests are used as a benchmark to compare with all future test results. Performance results obtained under the benchmark load should meet or exceed all indicated performance requirement s; otherwise tuning must begin with the benchmark load. Assuming no performance problems are noticed during this scenario, the results obtained can be used as “best case” results. These results indicate how the system performs when it is not under noticeable stress, but is still performing the required functions, thus allowing conclusions to be drawn about the performance of the system during higher load tests.

Sled will be benchmarked, in the environments described below. This benchmark is intended to provide a basis of comparison for future testing. Tuning may occur during the benchmarking effort if critical bottlenecks are detected.

Sled can then be re-benchmarked each time an iteration of either tuning or development has been completed on a module. This ensures that there is always a known valid point of comparison for all scheduled tests. The benchmark load will be 20 users, entering the system over a 30 minute period and performing the tasks outlines in section for a further hour i.e. total test time = 90 minutes.

Note: A subset of transactions from those completed within the UoL have been chosen for reporting purposes and indicate a good spread of the functionality. The appendices contain data for all transactions.

Benchmark Results 20 users

Transaction	Avg Time (sec)	Std Dev Time (sec)	95th Percentile Time (sec)
LOGIN	0.0	0.1	0.1
LOGIN_COURSE	0.1	0.1	0.2
ACTIVITY_MESSAGE_FACILITY	0.2	0.0	0.3
ACTIVITY_MESSAGE	0.2	0.0	0.3
SET_PROPERTY_SEND_MESSAGE	0.2	0.0	0.3
COMPLETE_ACTIVITY_VIEW_SAMPLE_MM_PROJ	0.2	0.0	0.2
ACTIVITY_UPLOAD_ACTIVITY	0.2	0.0	0.2
QTI_POST_SELECT_TITLE	0.2	0.0	0.2
ACTIVITY_COMPLETE_AND_UPLOAD_DIR_ACT	0.1	0.0	0.2

Table 2 – Benchmark Results (20 users)

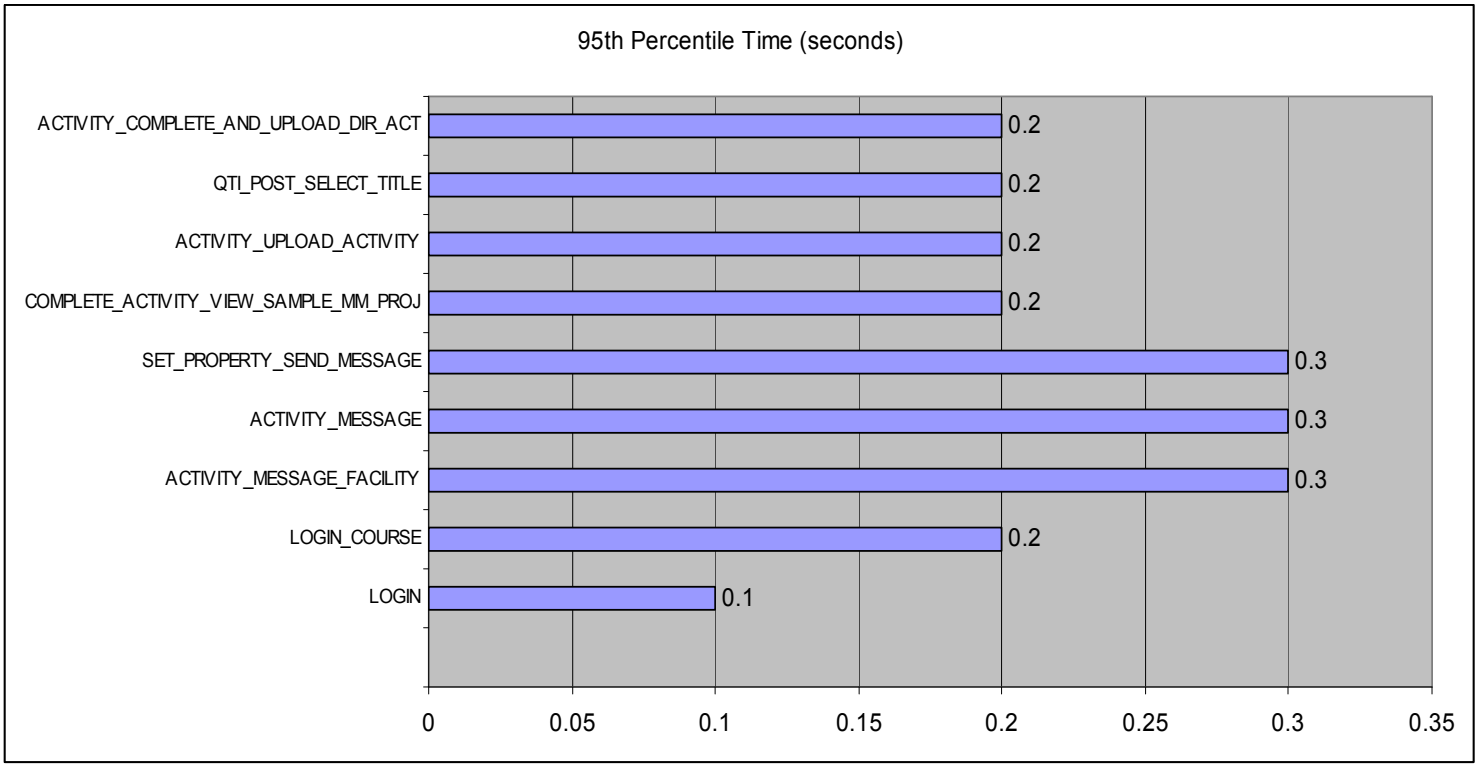

Figure 2 – Benchmark Chart (20 users)

Other scheduled test results

Scheduled Tests

The Execute scheduled tests aspect includes those activities that are mandatory to validate the performance of the system. They are

- Execute User Experience Tests
- Execute Stability Tests

User Experience Tests

User Experience Tests constitute what are considered to be expected real-world loads, from best case to worst case. Applying less than the expected worst-case load is useful in identifying major failings in a system, but does so in a way that doesn't highlight many of the more minor failings, allowing an easier analysis of results. When the load is equivalent to the expected real-world worst-case load, actual performance of the system can be measure and associated problems can be clearly identified.

These tests were designed to validate that the performance goals and requirements have been met. The results reported here represent the actual performance of the system upon conclusion of the Performance Testing effort.

The system was tested under loads of 50, 100, 150 and 200 virtual users.

User Experience Test Results

Virtual users were gradually released into the system over a 30 minute period. Once the ramp up period was completed, each scenario iterated several times for a total of an hour of relatively consistent load. User think times (time between user interactions with the system) were 30 seconds.

The page load times were measured in the same manner as they were under the Benchmark scenario to ensure consistency and validity between tests. Average times and 95th percentile times have been reported as well as standard deviations.

Note: A subset of transactions from those completed within the UoL have been chosen for reporting purposes and indicate a good spread of the functionality. The appendices contain data for all transactions.

50 User load

Transaction	Avg Time (sec)	Std Dev Time (sec)	95th Percentile Time (sec)
LOGIN	0.0	0.0	0.1
LOGIN_COURSE	0.2	0.1	0.3
ACTIVITY_MESSAGE_FACILITY	0.2	0.1	0.3
ACTIVITY_MESSAGE	0.2	0.1	0.3
SET_PROPERTY_SEND_MESSAGE	0.2	0.0	0.3
COMPLETE_ACTIVITY_VIEW_SAMPLE_MM_PROJ	0.2	0.1	0.3
ACTIVITY_UPLOAD_ACTIVITY	0.2	0.1	0.3
QTI_POST_SELECT_TITLE	0.2	0.1	0.3
ACTIVITY_COMPLETE_AND_UPLOAD_DIR_ACT	0.2	0.7	0.3

Table 3 – Experience Results (50 users)

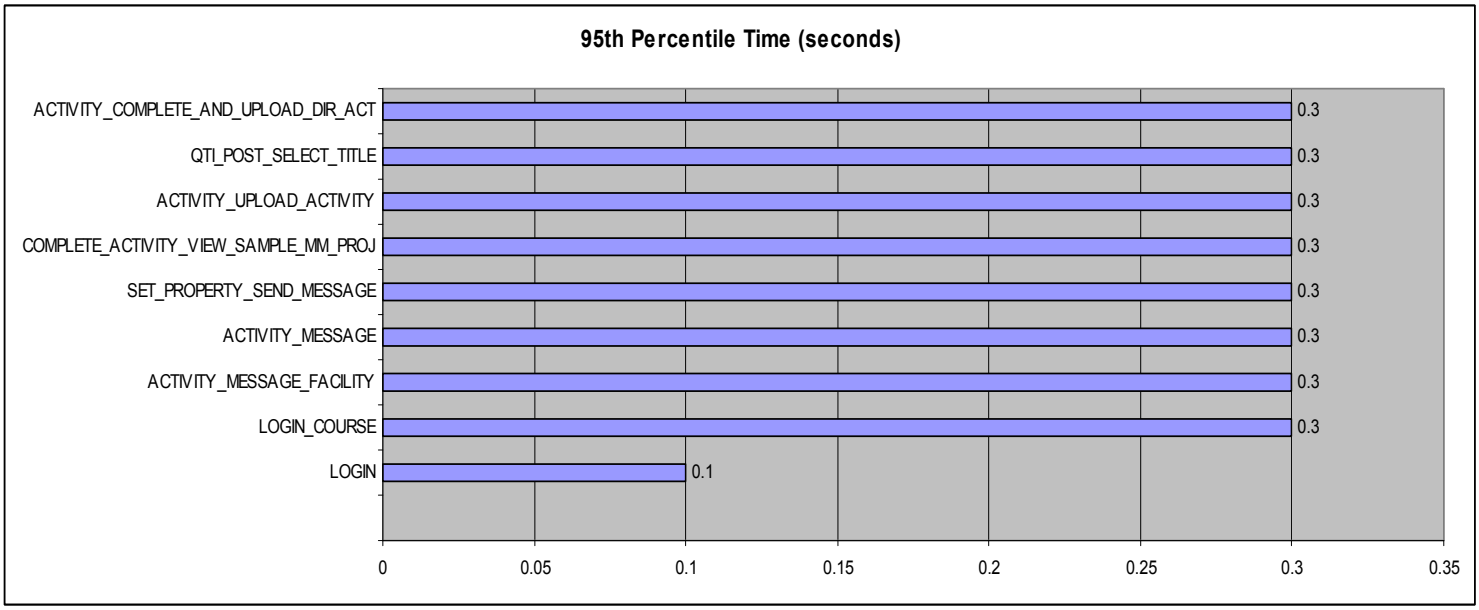

Figure 3 – Experience Chart (50 users)

100 User load

Transaction	Avg Time (sec)	Std Dev Time (sec)	95th Percentile Time (sec)
LOGIN	0.1	0.1	1.2
LOGIN_COURSE	0.2	0.2	0.5
ACTIVITY_MESSAGE_FACILITY	0.3	0.4	0.9
ACTIVITY_MESSAGE	0.2	0.2	0.4
SET_PROPERTY_SEND_MESSAGE	0.2	0.2	0.4
COMPLETE_ACTIVITY_VIEW_SAMPLE_MM_PROJ	0.3	0.3	0.8
ACTIVITY_UPLOAD_ACTIVITY	0.3	0.3	1.0
QTI_POST_SELECT_TITLE	0.3	0.4	0.9
ACTIVITY_COMPLETE_AND_UPLOAD_DIR_ACT	0.2	0.2	0.4

Table 4 – Experience Results (100 users)

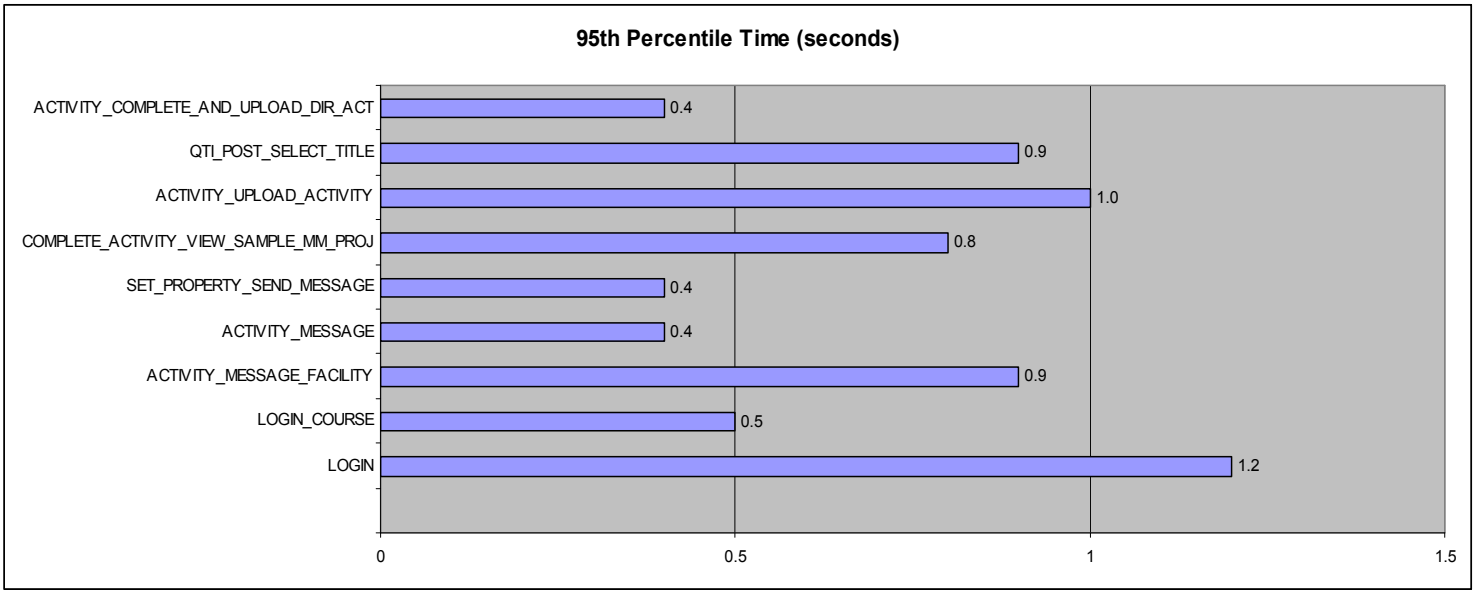

Figure 4 – Experience Chart (100 users)

150 User load (1.5X the acceptance load)

Transaction	Avg Time (sec)	Std Dev Time (sec)	95th Percentile Time (sec)
LOGIN	0.1	0.2	0.2
LOGIN_COURSE	0.9	0.8	2.4
ACTIVITY_MESSAGE_FACILITY	1.0	1.0	2.6
ACTIVITY_MESSAGE	0.9	1.0	2.8
SET_PROPERTY_SEND_MESSAGE	1.0	1.0	3.0
COMPLETE_ACTIVITY_VIEW_SAMPLE_MM_PROJ	1.5	1.1	3.6
ACTIVITY_UPLOAD_ACTIVITY	1.4	1.1	3.4
QTI_POST_SELECT_TITLE	1.4	1.1	3.4
ACTIVITY_COMPLETE_AND_UPLOAD_DIR_ACT	1.4	1.0	3.1

Table 5 – Experience Results (150 users)

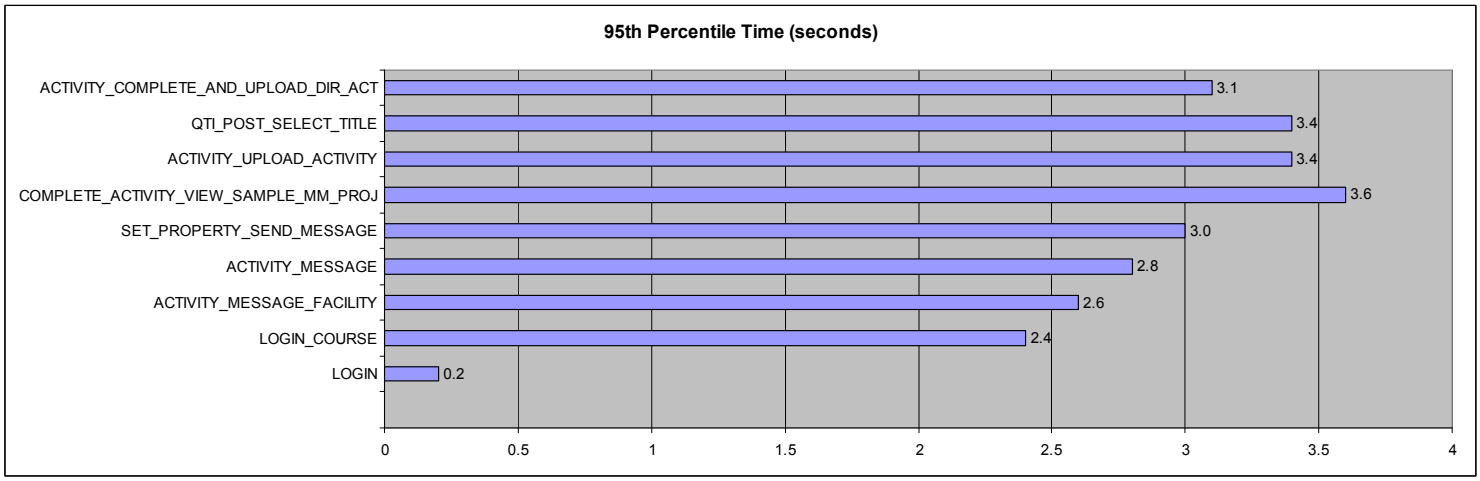

Figure 5 – Experience Chart (150 users)

200 User load (2X the acceptance load)

Transaction	Avg Time (sec)	Std Dev Time (sec)	95th Percentile Time (sec)
LOGIN	1.1	5.8	0.5
LOGIN_COURSE	15.8	17.9	44.1
ACTIVITY_MESSAGE_FACILITY	20.8	22.7	51.6
ACTIVITY_MESSAGE	19.6	21.3	48.3
SET_PROPERTY_SEND_MESSAGE	20.8	22.4	50.6
COMPLETE_ACTIVITY_VIEW_SAMPLE_MM_PROJ	32.4	19.3	50.1
ACTIVITY_UPLOAD_ACTIVITY	35.5	18.9	55.4
QTI_POST_SELECT_TITLE	40.0	17.3	55.8
ACTIVITY_COMPLETE_AND_UPLOAD_DIR_ACT	44.0	10.6	54.8

Table 6 – Experience Results (200 users)

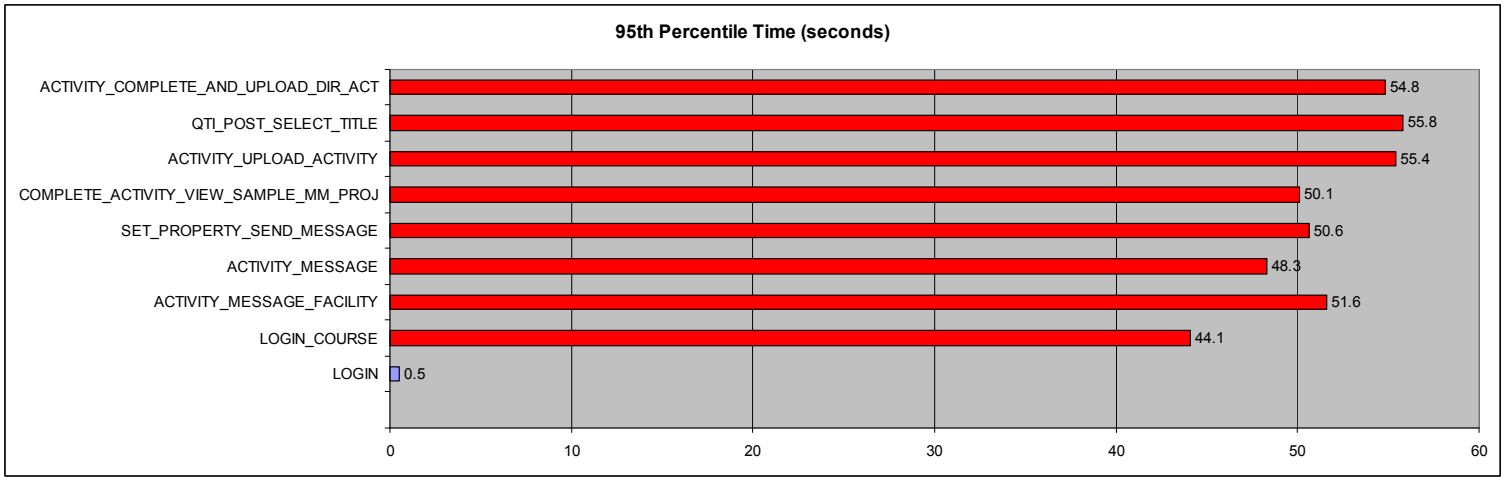

Figure 6 – Experience Chart (200 users)

Stability Tests

Stability scenarios test a system at and beyond the worse expected demand it is likely to face. The majority of critical deficiencies in the system will have already been identified during the execution of load tests, so this phase deals more with assessing the impact on performance and functionality under a heavy or reasonable load. Stability scenarios will also identify many other system bottlenecks not previously noticed, which may in fact be partially responsible for identified problems.

Heavy load scenarios are generally designed to be far more than a system can handle. They are used not to identify **if** a system fails, but **where** it fails first, how badly and why. By answering the *why* question, it can be determined whether a system is as stable as it needs to be.

Stress test

Stress tests are tests that use real-world distributions and user communities, but under extreme conditions. It is common to execute stress tests that are 150% of expected peak user load sustained over 12 hours with normal ramp up and ramp down time (users entering over a 30 minute period).

Stress Test Results

Transaction	Avg Time (sec)	Std Dev Time (sec)	95th Percentile Time (sec)
LOGIN	2.7	4.6	12.8
LOGIN_COURSE	7.1	8.7	22.7
ACTIVITY_MESSAGE_FACILITY	7.9	9.0	22.3
ACTIVITY_MESSAGE	7.0	7.8	19.5
SET_PROPERTY_SEND_MESSAGE	7.9	8.9	22.1
COMPLETE_ACTIVITY_VIEW_SAMPLE_MM_PROJ	5.6	7.7	21.0
ACTIVITY_UPLOAD_ACTIVITY	5.2	7.6	21.4
QTI_POST_SELECT_TITLE	8.7	9.6	24.0
ACTIVITY_COMPLETE_AND_UPLOAD_DIR_ACT	6.5	8.2	21.5

Table 7 – Stress Results (150 users)

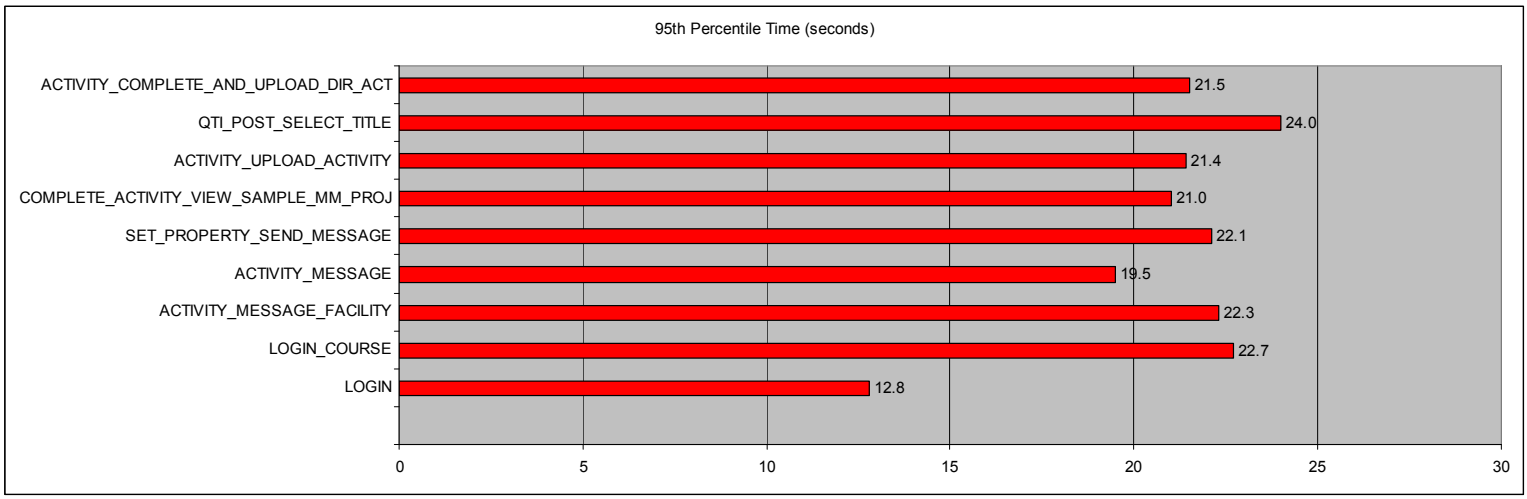

Figure 7 – Stress Chart (150 users)

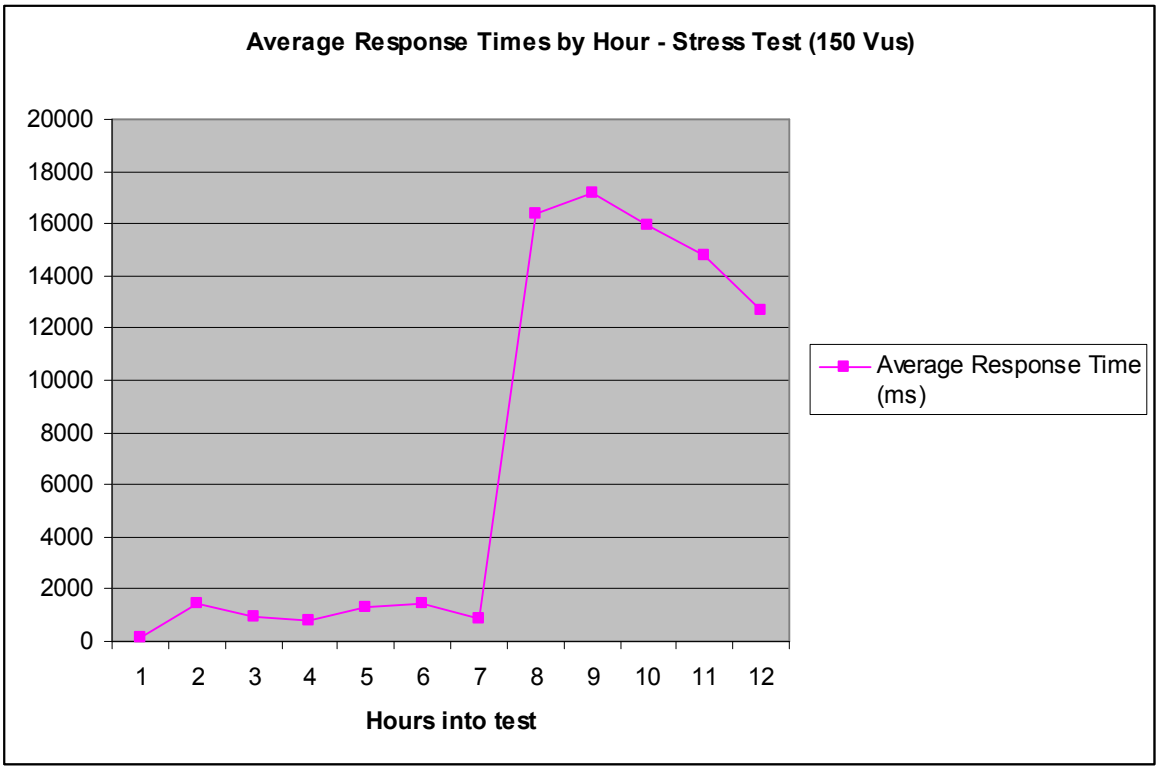

Figure 8 – Stress Chart Response times by hour (150 users)

Conclusions and recommendations

Consolidated Results

Summary Comparison					
Statistic/ Concurrent Users	**1**	**50**	**100**	**150**	**200**
Times Recorded	278	1070	2129	3118	2418
Times > Goal	0	0	0	0	1243
% Times > Goal	0%	0%	0%	0%	51.40%
Typical Average Time	0.1	0.2	0.2	1.1	25.6
Typical 95th Percentile Time	0.2	0.3	0.7	2.7	45.7

Conclusions

The goal of < 5 second response time for a target load of 100 users was met 100% of the time. The goal was also met 100% of the time at 1.5 X the target load i.e. 150 users. Even at twice the target load close to 50% of the response times were within the specified goal duration.

The results for 1, 50, 100 and 150 users show performance degrading within tight limits with the addition of a heavier user load. The 200 user load appears to have reached the limits of the test hardware in terms of response time however no exceptions were produced during the test. Based on the goals set in the performance acceptance criteria the system is ready for production use.

Appendix

Due to the amount of data generated by these tests the data is not reproduced here but can be downloaded in full at: